Adversarial Attacks in the Audio Domain

CS349 Machine Learning
Northwestern University
12.1.21

Patrick O'Reilly

github.com/oreillyp/adv_audio_intro
Adversarial Examples Fool Neural Networks
Adversarial Examples Fool Neural Networks

\[ x \]

“panda”
57.7% confidence

\[ + \ 0.007 \times \ sign(\nabla_x J(\theta, x, y)) \]

“nematode”
8.2% confidence

\[ = \]

\[ x + \epsilon \ sign(\nabla_x J(\theta, x, y)) \]

“gibbon”
99.3% confidence

(Goodfellow et al. 2014)
Adversarial Examples Fool Neural Networks
Adversarial Examples Fool Neural Networks

\[ x + \delta \]
Neural Networks Power Voice Interfaces

**Voice-based** machine-learning systems for authentication and control are common in products such as mobile devices, vehicles, and household appliances.
What Systems Might Attackers Target?
What Systems Might Attackers Target?

Recognize “Hey Alexa,” “OK Google,” “stop,” “go,” ...

Verify a speaker’s identity (against enrolled profile)

Transcribe all incoming speech

Wake-word detection, speech command recognition

Automatic speaker verification, speaker recognition

Automatic speech recognition
...Has Anyone Looked Into This?
...Has Anyone Looked Into This?
...Has Anyone Looked Into This?

~20-30 relevant attacks published, most since 2018
Has Anyone Looked Into This?

~20-30 relevant attacks published, most since 2018

A similar number of defenses have been proposed.
What Systems Might Attackers Target?

Recognize “Hey Alexa,” “OK Google,” “stop,” “go,” ...

Verify a speaker’s identity (against enrolled profile)

Transcribe all incoming speech

Wake-word detection, speech command recognition

Automatic speaker verification, speaker recognition

Automatic speech recognition
How Do We Make Adversarial Examples?
How Do We Make Adversarial Examples?

\[ f(x + \delta_{final}) = \text{BOB} \]
How Do We Make Adversarial Examples?

\[ f(x + \delta_{\text{final}}) = \text{BOB} \]
How Should We Attack?

NOT BOB ➔ BOB
Effective and Inconspicuous Over-the-Air Adversarial Examples with Adaptive Filtering

Patrick O’Reilly\textsuperscript{1}, Pranjal Awasthi\textsuperscript{2}, Aravindan Vijayaraghavan\textsuperscript{1}, Bryan Pardo\textsuperscript{1}

\textit{Submitted to ICASSP ‘22}

\textsuperscript{1}. Northwestern University
\textsuperscript{2}. Google Research

\url{interactiveaudiolab.github.io/project/audio-adversarial-examples.html}
How Should We Attack?

<table>
<thead>
<tr>
<th>Approach</th>
<th><strong>image-domain</strong> (sample-wise additive noise)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptual</td>
<td></td>
</tr>
<tr>
<td>Regularization</td>
<td><strong>simple</strong> ($L_2$ penalty)</td>
</tr>
<tr>
<td>Perceptual Quality</td>
<td><strong>poor</strong> (perturbation is obvious)</td>
</tr>
</tbody>
</table>
How Should We Attack?

*Qin et al. (2019), Szurley & Kolter (2019), Dörr et al. (2020), Wang et al. (2020)

**Approach**

<table>
<thead>
<tr>
<th>“Generic”</th>
<th>Qin et al.*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Perceptual Regularization</strong></td>
<td><strong>Perceptual Quality</strong></td>
</tr>
<tr>
<td>image-domain (sample-wise additive noise)</td>
<td>image-domain (sample-wise additive noise)</td>
</tr>
<tr>
<td>simple ($L_2$ penalty)</td>
<td>complex (frequency masking loss)</td>
</tr>
<tr>
<td>poor (perturbation is obvious)</td>
<td>good (perturbation is subtle)</td>
</tr>
</tbody>
</table>
How Should We Attack?

*Qin et al. (2019), Szurley & Kolter (2019), Dörr et al. (2020), Wang et al. (2020)

<table>
<thead>
<tr>
<th>Approach</th>
<th>“Generic”</th>
<th>Qin et al.*</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>image-domain (sample-wise additive noise)</td>
<td>image-domain (sample-wise additive noise)</td>
<td>audio-domain (adaptive filtering)</td>
</tr>
<tr>
<td>Perceptual</td>
<td>simple ($L_2$ penalty)</td>
<td>complex (frequency masking loss)</td>
<td>simple ($L_2$ penalty)</td>
</tr>
<tr>
<td>Regularization</td>
<td>poor (perturbation is obvious)</td>
<td>good (perturbation is subtle)</td>
<td>good (perturbation is subtle)</td>
</tr>
<tr>
<td>Perceptual Quality</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How Should We Attack?

*Qin et al. (2019), Szurley & Kolter (2019), Dörr et al. (2020), Wang et al. (2020)*

<table>
<thead>
<tr>
<th></th>
<th>“Generic”</th>
<th>Qin et al.*</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Approach</strong></td>
<td>❌</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Perceptual</strong></td>
<td>✔️</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Regularization</strong></td>
<td>✔️</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Perceptual Quality</strong></td>
<td>❌</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
How Should We Attack?

Qin et al.*

Proposed
Adversarial Examples Fool Neural Networks

\[ x \]

“panda”
57.7% confidence

\[ + .007 \times \]

\[ \text{sign}(\nabla_x J(\theta, x, y)) \]

“nematode”
8.2% confidence

\[ = \]

\[ x + \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \]

“gibbon”
99.3% confidence

(Goodfellow et al. 2014)
Let's Attack a Voice Interface
Let's Attack a Voice Interface: Pick a Task

Speaker Verification: confirm a speaker’s claimed identity (against enrolled profile)
Let's Attack a Voice Interface: Pick a Task

We want a large and accurate model, as in many applications (e.g. mobile banking) speaker verification models are deployed in the cloud rather than on-device.
Let's Attack a Voice Interface: Pick a Task

Specifically, we’ll use the ResNetSE34V2 model proposed by Heo et al. (2020), available at https://github.com/clovaai/voxceleb_trainer
Let's Attack a Voice Interface: Pick an Objective
Let's Attack a Voice Interface: Pick an Objective

Following Zhang et al. (2021), for the sake of simplicity we will attempt to spoof the embedding of a single utterance.
Let's Attack a Voice Interface: Pick a Setting

Over-the-line setting: the attack audio can be fed directly to the victim model over a purely digital channel.
Let's Attack a Voice Interface: Pick a Setting

**Over-the-line setting:** the attack audio can be fed directly to the victim model over a purely digital channel.

**Over-the-air setting:** malicious audio is played through a speaker and received by a microphone before entering the victim model.
Let's Attack a Voice Interface: Pick a Setting

**Over-the-line setting:** the attack audio can be fed directly to the victim model over a purely digital channel.

**Over-the-air setting:** malicious audio is played through a speaker and received by a microphone before entering the victim model.
Let's Attack a Voice Interface: Pick a Setting

**Over-the-line setting:** the attack audio can be fed directly to the victim model over a purely digital channel.

**Over-the-air setting:** malicious audio is played through a speaker and received by a microphone before entering the victim model.
Let's Attack a Voice Interface: System Design

Classify $(x + \delta)$ as $y$.

Make $\delta$ "imperceptible".
Let's Attack a Voice Interface: System Design

Adversarial Perturbation

Speech Source

Acoustic Distortions

Preprocessing

Model

Embedding

$L_{adv}(f, x, \delta, y)$

Classify $(x + \delta)$ as $y$

Make $\delta$ "imperceptible"

Inputs:
LibriSpeech test-other partition (4s)
Let's Attack a Voice Interface: System Design

$\text{Classify } (x + \delta) \text{ as } y$

Over-the-Air Simulation:
- time offset
- Gaussian noise
- environmental noise
- reverb
- bandpass filtering

Make $\delta$ “imperceptible”
Let's Attack a Voice Interface: System Design

Classify $(x + \delta)$ as $y$

Preprocessing:
- normalization
- voice activity detection (VAD)

Make $\delta$ “imperceptible”
Let's Attack a Voice Interface: System Design

Classify \((x + \delta)\) as \(y\)

Model: ResNetSE34V2

Make \(\delta\) “imperceptible”
Let's Attack a Voice Interface: System Design

\[ \nabla \mathcal{L}_{aux} \rightarrow \text{Adversarial Perturbation} \rightarrow \nabla \mathcal{L}_{adv} \]

\[ \mathcal{L}_{aux}(x, \delta) \]

Make \( \delta \) “imperceptible”

Classify \((x + \delta)\) as \(y\)
Let's Attack a Voice Interface: The Noise Issue

Let's attack a voice interface by adding noise to the input. The challenge is to make the perturbation imperceptible, even in noisy environments.

PROBLEM: challenging settings can induce noisy perturbations, even with a good auxiliary loss!

\[ \nabla \mathcal{L}_{aux} \rightarrow \text{Adversarial Perturbation} \rightarrow \nabla \mathcal{L}_{adv} \]

Make \( \delta \) "imperceptible"
Let's Attack a Voice Interface: Pick an Attack

Qin et al. (2019): speech recognition

Li et al. (2020): speaker recognition

Chen et al. (2020): speech recognition
Let's Attack a Voice Interface: System Design

\[ \nabla L_{aux} \]

Adversarial Perturbation

\[ \nabla L_{adv} \]

Make \( \delta \) “imperceptible”

Classify \((x + \delta)\) as \(y\)
Let's Attack a Voice Interface: Adaptive Filter Attack

\[ \nabla \mathcal{L}_{\text{aux}} \]

Adaptive Filter

\[ \nabla \mathcal{L}_{\text{adv}} \]

Speech Source

Acoustic Distortions

Preprocessing

Model

Embedding

\[ \mathcal{L}_{\text{adv}}(f, x, \delta, y) \]

Classify \((x + \delta)\) as \(y\)

Make \(\delta\) "imperceptible"
Adaptive Filters Let Us Shape Frequency Content
Adaptive Filters Let Us Shape Frequency Content

![Diagram of frequency content with amplitude and frequency in Hz]
Adaptive Filters Let Us Shape Frequency Content
Adaptive Filters Let Us Shape Frequency Content
Adaptive Filters Let Us Shape Frequency Content
Adaptive Filters Let Us Shape Frequency Content

Time Domain

Frequency Domain

1. Scale
2. Window, DFT
3. IDFT
4. Window, DFT
Adaptive Filters Let Us Shape Frequency Content

Time Domain

Frequency Domain

DFT

IDFT

element-wise multiply
Adaptive Filters Let Us Shape Frequency Content
Adaptive Filters Let Us Shape Frequency Content
Adaptive Filters Let Us Shape Frequency Content

$F$ frequency bands

$T$ frames
Adaptive Filters Let Us Shape Frequency Content

Filter Amplitudes (Unscaled)

- Frequency (Hz)
- Filter Amplitude
Adaptive Filters Let Us Shape Frequency Content
Let's Attack a Voice Interface: Adaptive Filter Attack

\[ \nabla \mathcal{L}_{aux} \]

Adaptive Filter

\[ \nabla \mathcal{L}_{adv} \]

Speech Source

Acoustic Distortions

Preprocessing

Model

Embedding

Cosine dist. (embeddings)

Classify \((x + \delta)\) as \(y\)

Make \(\delta\) “imperceptible”

NOT IMPORTANT
Let's Attack a Voice Interface: Adaptive Filter Attack

\[ \nabla \mathcal{L}_{aux} \]

\[ \nabla \mathcal{L}_{adv} \]

Optimize \( T \times F \) parameters

\((T \) frames and \( F \) frequency bands per frame)
Let's Attack a Voice Interface: Adaptive Filter Attack

Recall the iterative adversarial optimization procedure we discussed earlier.
Let's Attack a Voice Interface: Adaptive Filter Attack

Selective projected gradient descent (Bryniarski et al. 2021) - break up the updates

\[ \mathcal{L}_{adv} > 0 \quad \text{and} \quad \mathcal{L}_{adv} \leq 0 \]
Why Attack with Adaptive Filters?
Why Attack with Adaptive Filters?

1. Introducing perturbations at the filter representation, rather than the waveform, avoids noise-like artifacts
Why Attack with Adaptive Filters?

- "Generic" 89% effective
- Qin et al.* 93% effective
- Adaptive Filtering 95% effective
Why Attack with Adaptive Filters?

In general, when optimizing for more challenging distortions, attack success rate drops and artifacts become more audible.

“Generic”
86% effective

Qin et al.*
90% effective

Adaptive Filtering
93% effective
Why Attack with Adaptive Filters?

**User Study:** if we match effectiveness rates, listeners find our attack less conspicuous than Qin et al.* by a 2-to-1 margin.

- **“Generic”**
  - 89% effective

- **Qin et al.***
  - 93% effective

- **Adaptive Filtering**
  - 95% effective
Why Attack with Adaptive Filters?

<table>
<thead>
<tr>
<th>Waveform</th>
<th>$L_{\infty}$</th>
<th>Waveform</th>
<th>$L_2$</th>
<th>Perceptual Study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qin et al.*</td>
<td>0.08</td>
<td></td>
<td>1.97</td>
<td>34.1%</td>
</tr>
<tr>
<td>Adaptive Filtering</td>
<td>0.23</td>
<td></td>
<td>6.59</td>
<td>65.9%</td>
</tr>
</tbody>
</table>
### Why Attack with Adaptive Filters?

<table>
<thead>
<tr>
<th></th>
<th>Waveform $L_\infty$</th>
<th>Waveform $L_2$</th>
<th>Perceptual Study: Forced Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qin et al.*</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Adaptive Filtering</td>
<td>2.88x</td>
<td>3.35x</td>
<td>1.93x</td>
</tr>
</tbody>
</table>
Why Attack with Adaptive Filters?

2. When we use filters, we do not need a complex perceptual loss to produce inconspicuous attacks.
Why Attack with Adaptive Filters?

Two-stage frequency-masking attack: Qin et al. (2019), Szurley & Kolter (2019), Dörr et al. (2020), Wang et al. (2020)
Future Directions

Other recent works have also begun exploring attacks at representations other than the waveform (e.g. FoolHD, PhaseFool, Adversarial Music)
Future Directions

Other recent works have also begun exploring attacks at representations other than the waveform (e.g. FoolHD, PhaseFool, Adversarial Music)

We plan to explore filter-based attacks against more robust speaker verification pipelines, as well as other speech systems.
Future Directions

Other recent works have also begun exploring attacks at representations other than the waveform (e.g. FoolHD, PhaseFool, Adversarial Music)

We plan to explore filter-based attacks against more robust speaker verification pipelines, as well as other speech systems

We also plan to explore the implications of this work for improving the robustness of audio models against large-magnitude frequency-domain perturbations
Adversarial Attacks in the Audio Domain with Adaptive Filtering

Patrick O’Reilly¹, Pranjal Awasthi², Aravindan Vijayaraghavan¹, Bryan Pardo¹

https://interactiveaudiolab.github.io/project/audio-adversarial-examples.html

1. Northwestern University
2. Google Research
Thanks!